Straightforward intermediate rank tensor product smoothing in mixed models

Simon N. Wood[*], Fabian Scheipl[†], Julian J. Faraway[*]

January 6, 2012

Abstract

Tensor product smooths provide the natural way of representing smooth interaction terms in regression models because they are invariant to the units in which the covariates are measured, hence avoiding the need for arbitrary decisions about relative scaling of variables. They would also be the natural way to represent smooth interactions in mixed regression models, but for the fact that the tensor product constructions proposed to date are difficult or impossible to estimate using most standard mixed modelling software. This paper proposes a new approach to the construction of tensor product smooths, which allows the smooth to to be written as the sum of some fixed effects and some sets of i.i.d. Gaussian random effects: no previously published construction achieves this. Because of the simplicity of this random effects structure, our construction is useable with almost any flexible mixed modelling software, allowing smooth interaction terms to be readily incorporated into any Generalized Linear Mixed Model. To achieve the computationally convenient separation of smoothing penalties, the construction differs from previous tensor product approaches in the penalties used to control smoothness, but the penalties have the advantage over several alternative approaches of being explicitly interpretable in terms of function shape. Like all tensor product smoothing methods, our approach builds up smooth functions of several variables from *marginal* smooths of lower dimension, but unlike much of the previous literature we treat the general case in which the marginal smooths can be *any* quadratically penalized basis expansion, and there can be any number of them. We also point out that the imposition of identifiability constraints on smoothers requires more care in the mixed model setting than it would in a simple additive model setting, and show how to deal with the issue. An interesting side effect of our construction is that an ANOVA-decomposition of the

[*]Mathematical Sciences, University of Bath, Bath BA2 7AY U.K. s.wood@bath.ac.uk

[†]Department of Statistics, LMU, München, Germany

smooth can be read off from the estimates, although this is not our primary focus. We were motivated to undertake this work by applied problems in the analysis of abundance survey data, and two examples of this are presented.

Keywords: tensor product, smooth, smoothing spline ANOVA, low rank, space-time, spatio-temporal, identifiability constraint, mixed model.

1 Introduction

Generalized additive mixed models (GAMM, Lin and Zhang, 1999) combine the flexible modelling of the relationship between a response and predictors embodied in generalized additive models (GAM, Hastie and Tibshirani, 1986), with the flexible models of stochastic variability in the response provided by generalized linear mixed models (GLMM). Depending on the applied problem to hand, one can view GAMMs as adding random effects to GAMs, or as adding flexible fixed effects modelling to GLMMs, and the estimation strategies for GAMMs divide along similar lines. If the primary interest is in estimating the smooth relationships between the response and predictors, and the random effects structure is simple and low dimensional, then it is usually best to estimate the GAMM using methods designed for GAMs, treating the random effects in the same way that the smooth functions are treated (see Wood, 2008, 2011). Alternatively, if the random effects structure is richer and high dimensional, then GAM specific methods are usually inefficient or impractical, and it is better to represent the smooth functions as random effects, and estimate using methods designed for GLMMs.

The duality between spline like smooths and random effects that underpins these two strategies goes back to Kimeldorf and Wahba (1970) but straightforward methods for estimating smooths as mixed model components in GLMMs had to wait for the simple Pspline approach of Ruppert, Wand and Carrol (2003, see also Verbayla et al. 1999 and Eilers, 1999). They proposed representing 1D functions in mixed models using simple truncated power basis splines, with a ridge penalty. Such splines can be estimated as i.i.d. Gaussian random effects, rendering estimation straightforward with most flexible mixed modelling software. It was quickly realized that a simple re-parameterization trick would allow the same approach to be taken with any spline like smooth representable with a linear basis expansion and a quadratic penalty (e.g. Wood, 2004, Fahrmeir et al., 2004).

Unfortunately, to date the important class of *tensor product* smoothers can not be treated in this way. Tensor product smooths are the method of choice for representing smooth interaction terms in models (smooth functions of more than one variable), when the variables are represented in different units. The

idea is that when the relative scaling of variables is arbitrary then the smooth should be invariant to that scaling, and free from arbitrary decisions about the relative importance of smoothness with respect to those variables (for example, invariance should not be obtained by the artificial device of applying an artificial rescaling to the data, if poor smoother performance is to be avoided).

There is an extensive literature on tensor product smoothing, with Wahba (1990) and Gu (2002) providing good overviews of the full smoothing spline approach. Kim and Gu (2004), Eilers and Marx (2003), Wood (2006a), Belitz and Lang (2008) and Lee and Durban (2011) are among the papers discussing computationally efficient low rank tensor product smoothers, which offer feasible computation even with large data sets. However, no published approach to tensor product smoothing provides invariant smooths which can be represented as fixed effects plus a sequence of sets of i.i.d. Gaussian random effects, in the way that is needed for computation with most mixed modelling software. Instead all the published tensor product constructions result in mixed model representations in which at least one random effect covariance matrix has a non-standard form involving at least two variance parameters. For this reason the existing approaches require specialist software to be written in order to fit them (even Wood, 2006a, which does manage estimation via R package `nlme`, required the use of complex bespoke covariance classes).

This paper provides the first tensor product construction method which results in invariant smooths and has a mixed model representation involving no more than simple i.i.d. Gaussian random effects. We can not achieve this by simply re-writing existing constructions in some clever way, but must instead use a different set of smoothing penalties to those employed by previous authors (although some of the set will correspond to those employed by Gu, 2002, when the marginal penalties coincide). However, these penalties are directly interpretable in terms of function shape, which is a further advantage over several published alternatives, where the penalty meaning is not explicit.

2 The model and its representation

The general model class considered is

$$g(\mu_i) = \mathbf{X}_i\boldsymbol{\alpha} + \sum_j L_{ij} f_j + \mathbf{Z}_i\mathbf{b}, \quad \mathbf{b} \sim N(\mathbf{0}, \boldsymbol{\psi}_\theta), \quad y_i \sim \mathrm{EF}(\mu_i, \phi) \tag{1}$$

where the y_i are independent observations of a univariate response variable from an exponential family distribution with mean μ_i and scale parameter ϕ. g is a known smooth monotonic link function, $\mathbf{X}$ is a model matrix (the notation $\mathbf{A}_i$ denotes the i^{th} row of $\mathbf{A}$), $\boldsymbol{\alpha}$ is a vector of unknown parameters, $\mathbf{Z}$

is a model matrix for random effects $\mathbf{b}$, which have covariance matrix $\boldsymbol{\psi}_\theta$ parameterized by unknown parameter vector $\boldsymbol{\theta}$. L_{ij} is a known linear functional and f_j an unknown smooth function of one or more variables, with an unknown degree of smoothness. Associated with each f_j is one or more penalties measuring departure from smoothness, $J_j(f)$. Often the L_{ij} are simply evaluation functionals so that $L_{ij}f_j = f_j(x_{ji})$, but other common examples are the 'varying coefficient' term $L_{ij}f_j = f_j(x_{ji})z_{ji}$ where z_{ji} is a known covariate, or the 'signal regression' term $L_{ij}f_j = \int k_i(x)f_j(x)dx$, where $k_i(x)$ is an observed function.

Representing the f_j in (1) by intermediate rank penalized regression splines (e.g. Wahba, 1980, Parker and Rice, 1985, Eilers and Marx, 1996) results in a computationally convenient inferential framework for these models (e.g. Wood, 2004, 2008), particularly if each f_j is subject to only one smoothing penalty. Standard mixed modeling software can be used to estimate (1) in this case, which allows the models to employ rich random effects structures. However, if the f_j are functions of several variables, then single penalties usually arise only when it is appropriate to smooth isotropically, and isotropic smoothing is not appropriate for most interaction terms. For example, it is rarely appropriate to treat space and time isotropically when smoothing: indeed variables measured in different units should seldom be treated isotropically, and it is rare that there is a natural relative scaling of variables with different units that is apparent to the modeller *a priori*.

The key property of properly constructed smooth interaction terms is that they should be invariant to the relative scaling of their covariates when this scaling is arbitrary, and should achieve this invariance without arbitrary assumptions about the relative importance of smoothness with respect to these different covariates: this is the motivation underlying tensor product smoothing. In such smooths invariance is achieved by employing tensor products of spline bases to represent smooths, with each smooth subject to *multiple* smoothing penalties. The original work in this area employed full spline smoothers (see Wahba, 1990 and Gu, 2002 for overviews), but recent work has developed more computationally efficient approaches based on penalized regression splines (Gu and Kim, 2002 and Kim and Gu, 2004, Eilers and Marx, 2003, Wood 2006a, Belitz and Lang, 2008, and Lee and Durbán, 2011), which allows feasible computation with much larger datasets. The problem, in practice, is that multiple penalization makes it difficult or impossible to estimate these interaction smooths using most standard mixed modeling software (such as R package `lme4` or the procedures provided by SAS), substantially restricting the class of regression models in which they can be incorporated.

The purpose of this paper is to provide interaction smooths that *can* conveniently be estimated by

modern mixed modeling software. This is achieved via a construction method that allows tensor product smooths to be decomposed into components each subject to at most one penalty. The construction is fully automatic, and unlike most previous work on low rank tensor product smoothing, it is general, rather than focusing on particular marginal bases: this has the immediate benefit of allowing production of a rather natural three dimensional space-time smoothers constructed from a two dimensional thin plate spline for space, and a one dimensional spline for time. The construction results in a somewhat different set of penalties to those used by previous authors, but this has the advantage (shared by the full smoothing splines) that the penalties can have explicit interpretations in terms of function shape.

With such a general method, the modeler is free to specify the tensor product smooths best suited to the task at hand and to use the best estimation software available, rather than being restricted to particular smooths for which methods exist and to software that can cope with the construction.

2.1 Basis penalty smooths

First consider representing smooth functions that are univariate, or where isotropic smoothness is appropriate. The j^{th} smooth in (1) can be represented as

$$f_j(x) = \sum_k \beta_k b_{jk}(x)$$

where the β_k are unknown parameters and the $b_{jk}(x)$ are some known spline basis functions (the β_k are specific to f_j, here, but to avoid clutter we have not denoted this notationally). For spline like smoothers we can always write the penalty for f_j as $J_j(f_j) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_j\boldsymbol{\beta}$ where $\mathbf{S}_j$ is a positive semi-definite matrix of fixed coefficients. Taking a Bayesian perspective (e.g. Silverman, 1985) the penalties can be used to define (independent) improper priors on the wiggliness of each f_j, namely

$$\pi(\boldsymbol{\beta}) \propto \exp\left(-\lambda_j \boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_j\boldsymbol{\beta}/(2\phi)\right) \tag{2}$$

where the λ_j control the dispersion of the priors, and hence the smoothness of the f_j. Given $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, the MAP (maximum a posteriori) estimates/predictions for $\boldsymbol{\alpha}$, $\mathbf{b}$ and the spline coefficients are easily obtained by penalized likelihood maximization. An empirical Bayes approach can be used to estimate ϕ, $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ by marginal likelihood maximization after approximately integrating out the random components of $\mathbf{b}$, $\boldsymbol{\alpha}$ and the spline coefficients from their joint density with $\mathbf{y}$.

Computationally, the preceding estimation strategy can be achieved by representing (1) as a generalized linear mixed model, and estimating its variance components by Maximum Likelihood or REML. This representation is achieved by reparameterizing each f_j so that some of its basis functions represent

only the space of functions for which $J_j(f) = 0$ (the penalty null space), while the remainder represent the space of functions for which $J_j(f) > 0$ unless $f = 0$. The coefficients for the penalty null space of each f_j are treated as fixed effects, while the remaining coefficients are treated as random effects (they now have a *proper* distribution). Note that this is a computational trick to compute Bayesian estimates: it is very rare that the modeler really believes that the f_j are random functions re-drawn from (2) on each replication of the data, so the model is not really a frequentist mixed model. Estimating the model in this way is particularly appealing if the model has a relatively rich random effects structure in addition to the smooth components, but is only possible if mixed model estimation methods can be coerced into fitting with the random effects covariance structure implied by the function penalties, something which is straightforward for singly penalized smooths (e.g. Wood, 2004), but not otherwise.

2.2 Tensor product smooth bases

Now consider tensor product interaction terms, the main subject of this paper. The construction of a tensor product spline basis is best illustrated by considering a smooth of two variables, x and t, say. Start by representing smooth functions of x using the basis expansion

$$f(x) = \sum_k \alpha_k a_k(x)$$

where a_k is a known basis function and α_k a coefficient. A smooth function of x and t can be obtained by allowing $f(x)$ to vary smoothly with t. To achieve that we allow each coefficient α_k to vary smoothly with t by using a second basis expansion,

$$\alpha_k(t) = \sum_j \beta_{kj} b_j(t),$$

where the b_j are known basis functions and the β_{kj} are coefficients. So we now have a smooth function of x and t...

$$f(x,t) = \sum_{kj} \beta_{kj} b_j(t) a_k(x).$$

The b_j and a_k are the *marginal* basis functions for $f(x,t)$. Provided that the marginal bases are invariant, in the sense that any linear rescaling of x and t can be exactly compensated for by appropriate modification of α_k or β_j, then $f(x,t)$ is invariant to the relative scaling of x and t. Such a tensor product construction can be generalized to any number of variables, and x and/or t may themselves be vector valued (perhaps treated isotropically).

2.3 Scaling invariance in detail

Tensor product smooths are appropriate for smoothing with respect to multiple variables, when we don't know, a priori, how much to weight smoothness with respect to different variables. This situation applies particularly when the variables are measured in different units, so that there is no natural way to put them 'on the same scale' for smoothing. If f is a smooth function of several variables, then we can formulate a scaling invariance principle:

> Inference about f should not depend on arbitrary decisions about the relative scaling of variables, or about the relative penalization of variability of f with respect to those variables.

In other words we should make no assumption about how variability with respect to one variable should be weighted relative to variability with respect to another variable when judging smoothness: instead the relative weighting should be estimated from the data. In the degenerate case of no smoothing penalties, tensor product bases are scaling invariant in this sense, but penalties for tensor product smoothing must be constructed with some care to ensure that the principle is still satisfied under penalization.

Three examples of smoothing with respect to distance x, and time t serve to illustrate violations of scaling invariance:

1. Thin plate spline spline smoothing with respect to x and t is not scaling invariant, since, for example, different results will be obtained from the same data if we use units of mm and hours, as opposed metres and seconds. This is because the TPS penalty penalizes variability of f per unit change in x similarly to variability of f per unit change in t, irrespective of what the units are. Since there is no 'natural' choice for the units the results depend on an arbitrary choice.

2. We could scale x and t so that they lie in the unit square and then smooth with a TPS of LOESS smoother. Results will not then depend on the units of measurement, but the smooth is still not scaling invariant. This is because the relative penalization of variation of f w.r.t. x and t is now controlled arbitrarily by the range of the variables. For example, if the x, t domain is the unit square then we will penalize variability per unit change in x and t equally, while if the domain is a 10×1 rectangle then we will penalize variability per unit change in x much more heavily than variability per unit change in t. That is the relative penalization has been chosen arbitrarily.

3. We could smooth using a rectangular grid of tensor products of B-spline basis functions, with a penalty obtained by summing squared second differences of spline coefficients along the rows

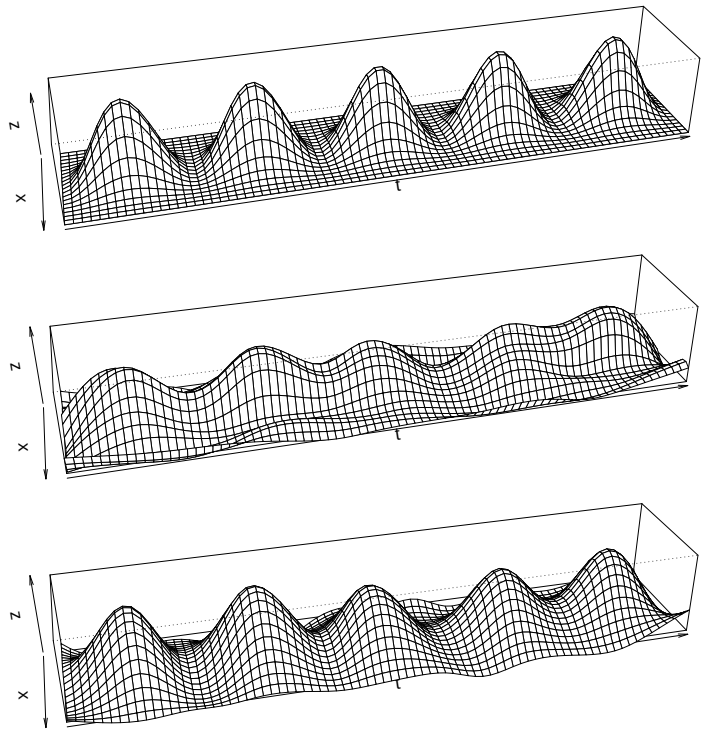

Figure 1: Illustration of lack of invariance in the P-spline smooth example 3 from section 2.3, which superficially appears invariant. **Top**: a true function of $x \in [0, 1]$ and $t \in [0, 5]$. The function was sampled noisily at 1000 random locations. **Middle**: a reconstruction from the data using a 20×20 tensor product of P-splines, with a single discrete penalty constructed by differencing coefficients along rows and columns, and smoothing parameter chosen by REML. Superficially the smooth appears invariant, since penalty and basis are unchanged under linear transformation of x and t. However, since the penalty contains no explicit information about the range of x and t, it actually penalizes variation per unit change in t much more heavily than variation per unit change in x, something which is essentially arbitrary, and results in a very poor reconstruction **Lower**: exactly the same data were supplemented by 20 extra data points with $y = 0$, $t = 2.5$ and x evenly spaced between -4 and 0, *which were given zero weight in fitting* and the same smoother was applied again. The improvement occurs because the penalty for the new data now happens to weight variation with respect to x and t equally, as a result of the change in covariate range. The large differences between the plots caused by inclusion of uninformative data are clearly undesirable, and do not occur if the same experiment is repeated with a properly invariant smoother.

and columns of the grid. Since neither penalty, nor basis, appear to depend on the units of x and t this smoother superficially appears scaling invariant, but in reality the relative penalization of variability per unit change in x and t are implicitly set by the discrete penalty and the range of the covariates. As in example 2 this relative penalization has been selected arbitrarily. Figure 1 illustrates the sensitivity of this smoother to the range of the data, emphasising that merely ignoring the relative scaling of covariates in the formulation of a smoother does not make the smoother invariant to that relative scaling in any useful sense.

Appendix 0 shows that it is straightforward to obtain sufficient conditions for scaling invariance. Let $\mathbf{f}$ be the vector of values of the smooth evaluated at the covariate values. In general this can be expressed as

$$\mathbf{f} = \mathbf{X}\boldsymbol{\beta} + \sum_j \mathbf{Z}_j \mathbf{b}_j$$

where the $\mathbf{Z}_j$ and $\mathbf{X}$ are matrices of evaluated basis functions and the vectors $\boldsymbol{\beta}$ and $\mathbf{b}_j$ contain coefficients. Associated with each coefficient vector $\mathbf{b}_j$ is a penalty $\sum_k \lambda_{jk} \mathbf{b}_j^{\mathrm{T}} \mathbf{S}_{jk} \mathbf{b}_j$. Without loss of generality we can assume that the smooth has been constructed in scale dependent form, so that we have made explicit the dependence on covariate scale of the measure of function smoothness used for penalization (this is automatic for derivative penalties, and for P-spline penalties simply means dividing coefficient differences by the corresponding knot spacing). Then the smooth will be scaling invariant if

SI1 The only basis change caused by linear rescaling of the covariates of the smooth is that the $\mathbf{Z}_j$ and each column of $\mathbf{X}$ may each be multiplied by it own constant.

SI2 The only change to the smoothing penalties occasioned by linear rescaling of the covariates is that each $\mathbf{S}_{jk}$ may be multiplied by its own constant.

Example 1, above, fails to meet these conditions immediately, and 2 and 3 fail as soon as the penalties are re-written to make the dependence on scale explicit, rather than implicit. In contrast, for example, the tensor product smooth construction discussed in Wood, 2006a, satisfies the conditions.

2.4 Previous approaches, and what is new here

In practice, for invariance of the tensor product basis of section 2.2 to result in scaling invariant estimates when smoothing, it is necessary for $f(x, t)$ to be subject to multiple smoothing penalties (at least if the penalties are to do any useful degree of smoothing). Multiply penalized tensor product smooths have

been used in the full smoothing spline literature for some time (e.g. Wahba, 1990, Gu, 2002), but the first use in penalized regression spline smoothing seems to be Eilers and Marx (2003), who used double penalization of a tensor product of P-splines (Eilers and Marx, 1996). Wood (2006a) generalized their construction to use any marginal smooth defined by a basis expansion and quadratic penalty, modifying the penalty construction method a little to improve interpretability, and also discussing ANOVA decompositions of functions. Belitz and Lang (2008) and Lee and Durbán (2011) use different penalties again in more detailed studies of functional ANOVA in the setting of Eilers and Marx (1996) P-splines. Notice that all the penalized regression spline approaches mentioned above result in the same tensor product space, given the same marginals: it is only their different penalties that distinguish them. However, for all these approaches it is difficult to use the resulting smooths as mixed model components to be estimated with standard mixed modeling software, because many coefficients of the smooth are subject to multiple penalties (equivalently, penalties with multiple smoothing parameters). Wood (2006a) did manage to estimate such models using Penalized Quasi-Likelihood (PQL, Breslow and Clayton, 1993) and R package `nlme`, but this involved complex code that relied heavily on the inner workings of `nlme` (Pinheiro and Bates, 2000). Furthermore, PQL is known to be poor for binary and low count data, unlike the more modern methods used in `lme4` (Bates and Maechler, 2010) or SAS, for example.

In this paper we propose a novel tensor product construction which produces low rank tensor product smoothers which are scaling invariant, can be constructed from any marginal smoothers defined by a quadratically penalized basis expansion, have penalties that are interpretable in terms of function shape and have coefficients that are each penalized by at most one penalty (which is linear in one unknown smoothing parameter). It is the latter feature which is novel and distinguishes our method from the published alternatives. In short we provide a general recipe for incorporating tensor product smooths into GLMMs estimable with the best modern software.

2.5 Marginal smooth reparameterization

The general tensor product construction to be presented in section 3 requires a reparameterization of each marginal of the tensor product smooth, so that its penalty has a simple ridge form penalizing only some marginal coefficients. This reparameterization is covered here. (It is a trivial consequence of the tensor product construction that the function space of the tensor product smooth is invariant to any invertible linear reparameterization of its marginals.)

Consider a marginal smooth $f(x)$ (where x may itself be a vector quantity) with a representation in

terms of known basis functions $b_k(x)$ and unknown coefficients β_k,

$$f(x) = \sum_k \beta_k b_k(x).$$

It will be subject to a single penalty term $\lambda \boldsymbol{\beta}^{\mathrm{T}} \mathbf{S} \boldsymbol{\beta}$, where λ is an unknown smoothing parameter, and $\mathbf{S}$ is a known positive semi-definite matrix (the *marginal penalty matrix*). Suppose that we have observations relating to f at $x_1, x_2, \ldots$. Then $[f(x_1), f(x_2), \ldots]^{\mathrm{T}} = \mathbf{f} = \boldsymbol{\mathcal{X}} \boldsymbol{\beta}$, where $\mathcal{X}_{ij} = b_j(x_i)$, and $\boldsymbol{\mathcal{X}}$ is the *marginal model matrix*.

The following general reparameterization is useful. Form the symmetric eigendecomposition $\mathbf{S} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^{\mathrm{T}}$, where the eigenvalues of $\mathbf{S}$ are arranged in order of decreasing magnitude down the leading diagonal of $\boldsymbol{\Lambda}$. The last M eigenvalues will be zero, where M is the dimension of the space of functions for which $\boldsymbol{\beta}^{\mathrm{T}} \mathbf{S} \boldsymbol{\beta} = 0$. Let $\bar{\boldsymbol{\Lambda}}$ be the diagonal matrix such that $\bar{\Lambda}_{ii} = \sqrt{\Lambda_{ii}}$ if $\Lambda_{ii} > 0$ and $\bar{\Lambda}_{ii} = 1$ if $\Lambda_{ii} = 0$. Now reparameterize so that $\boldsymbol{\beta}' = \bar{\boldsymbol{\Lambda}} \mathbf{U}^{\mathrm{T}} \boldsymbol{\beta}$, the model matrix becomes $\boldsymbol{\mathcal{X}}' = \boldsymbol{\mathcal{X}} \mathbf{U} \bar{\boldsymbol{\Lambda}}^{-1}$, and the penalty matrix is an identity matrix with the last M elements on the leading diagonal zeroed.

The penalty structure means that the last M elements of $\boldsymbol{\beta}'$ are unpenalized, while the remaining components are subject to a ridge penalty. This suggests rewriting the smooth as

$$\mathbf{f} = \mathbf{X} \boldsymbol{\delta} + \mathbf{Z} \mathbf{b}$$

where $\mathbf{X}$ is the last M columns of $\boldsymbol{\mathcal{X}}'$ and $\mathbf{Z}$ is the other columns. Correspondingly, $\boldsymbol{\delta}$ is the last M elements of $\boldsymbol{\beta}'$ and $\mathbf{b}$ is the other elements. Then the penalty on the smooth becomes the simple ridge penalty $\lambda \mathbf{b}^{\mathrm{T}} \mathbf{b}$. When estimating the smooth by mixed model methods, the $\boldsymbol{\delta}$ are treated as fixed effects parameters, and the $\mathbf{b}$ as i.i.d. Gaussian random effects.

The reparameterization employed so far (which is from Wood, 2004) does not guarantee that the constant function is one of the null space basis functions. This is a deficiency if functional ANOVA decompositions are of interest (see Gu 2002 and Lee and Durbán, 2011). When a basis for the null space which explicitly includes the constant function is known, then this could simply be used in place of the automatically generated basis. However not all smooths have such a known null space basis, and we are interested in providing a method that works in general, so we propose using the following fully automatic null space reparameterization.

If g is a function in the penalty null space then $\mathcal{P}_N = \sum_i (g(x_i) - \bar{g})^2$, further shrinks g towards the space of constant functions. Defining $\mathbf{D} = \mathbf{X} - \mathbf{1}\mathbf{1}^{\mathrm{T}} \mathbf{X}/n$ then $\mathcal{P}_N = \boldsymbol{\delta}^{\mathrm{T}} \mathbf{D}^{\mathrm{T}} \mathbf{D} \boldsymbol{\delta}$. Proceeding exactly as was done with the penalty matrix $\mathbf{S}$, above, form the eigendecomposition

$$\mathbf{D}^{\mathrm{T}} \mathbf{D} = \boldsymbol{\mathcal{U}} \boldsymbol{\Omega} \boldsymbol{\mathcal{U}}^{\mathrm{T}}$$

and reparameterise so that the null space model matrix is now $\mathbf{X}\boldsymbol{\mathcal{U}}$. Provided that the null space of $\mathbf{S}$ includes the constant function in its span, then the last column of $\mathbf{X}\boldsymbol{\mathcal{U}}$ will be a column of constants corresponding to the null space of $\mathcal{P}_N$. This approach works even when the null space basis has no simple known form (e.g. a high order Markov random field).

3 A general construction of tensor product smooths

We propose to create tensor product smooths from any combination of marginal smooths with a linear basis expansion and quadratic penalty, by first reparameterizing each smooth as in section 2.5. After reparameterization, the basic tensor product basis construction of section 2.2 is applied in such a way that the columns of the tensor product smooth model matrix are divided into non-overlapping subsets. Borrowing directly from smoothing spline ANOVA (in particular see Wahba, 1990, section 10.2 or Gu 2002, section 2.4), each subset is constructed from the product of some null space basis functions for some margins and some range space basis functions for other margins (including the all null spaces and all range spaces products). The coefficients for the subset constructed from the product of all marginal null spaces is unpenalized. The coefficients for all other subsets are each subject to a subset specific ridge penalty with a single smoothing parameter.

In the following let $[\mathbf{X}^j]$ denote the columns of $\mathbf{X}^j$, treated as separate elements of a set. The construction proceeds as follows:

1. Start with a set of d marginal smooths.
2. Re-parameterize the smooths as in section 2.5, so that they each have a fixed model matrix $\mathbf{X}^j$ a random effects model matrix $\mathbf{Z}^j$ and a penalty matrix $\mathbf{I}$ (associated with random effects only).
3. Create a set $\gamma = \{\mathbf{X}^1, \mathbf{Z}^1\}$ (or $\gamma = \{[\mathbf{X}^1], \mathbf{Z}^1\}$).
4. Repeat steps 5-7, for i in 2 to d...
5. Form row-wise Kronecker products of $\mathbf{X}^i$ (or of all the columns $[\mathbf{X}^i]$) with all elements of γ.
6. Form row-wise Kronecker products of $\mathbf{Z}^i$ with all elements of γ.
7. Append the results of the previous two steps to the set γ.

8. The model matrix for the tensor product smooth is given by appending all the elements of γ, columnwise. Each element of γ has an associated identity penalty, except for the element(s) which involve(s) no $\mathbf{Z}^j$ term, which is unpenalized.

9. For numerical stability, each element of γ can be scaled to have unit (Frobenius) norm.

10. Any identifiability constraint must be applied in such a way that the non-overlapping diagonal structure of the penalties is maintained: applying constraints only to the unpenalized element(s) of γ is usually the easiest way to ensure this.

The alternatives in parentheses at steps 3 and 5 ensure strict invariance by treating each basis function of each penalty null space separately in the construction. This results in more penalties than treating the null space as a whole. When the null space terms are treated whole, then sensitivity to covariate scaling can be artificially avoided by scaling of the columns of each $\mathbf{X}^j$ to have the same norm: this does not achieve full scaling invariance, however.

The above construction creates a tensor product basis, where each coefficient is subject to at most one simple ridge penalty. Hence the the representation of the smooths as random effects terms with i.i.d. normal coefficients is straightforward. Appendix 1 provides a specific example of how the general construction works in detail, for two example marginal bases.

Notice that our general construction creates a basis spanning the same space as previous constructions, given the same marginals. Indeed the exact basis construction has been available in R package `mgcv` for tensor products of thin plate splines since 2005 (e.g. `te(x,z,bs="tp",np=FALSE)` in a `gam` formula gives such a basis), and it is the construction used for tensor products of 2nd order P-splines in Lee and Durbán (2011). The innovation here is the penalties, which are different from those used previously, achieving separation, invariance and explicit representation in terms of function shape.

Now consider why the construction achieves scaling invariance, and the explicit form of the induced penalties.

3.1 Scaling invariance of the smooths

Our construction results in scaling invariant smooths if the marginal smooths are scaling invariant, in that they meet our sufficient conditions for scaling invariance from section 2.3. This is easy to show. Under section 2.5 re-parameterization the dependence of the penalty on covariate scale is transferred from the penalty matrices to the marginal smoothing bases, so that re-parameterization does not alter the penalty

matrices in our construction, and SI2 is satisfied.

Let $\mathbf{X}_j$ denote a submatrix of the tensor product smooth model matrix with corresponding coefficients all penalized by the same ridge penalty, with smoothing parameter λ_j. SI1 is that, for any j, linear rescaling of covariates leads only to a multiplicative change in $\mathbf{X}_j$. This occurs if its component $\mathbf{Z}^i$ and $\mathbf{X}^i$ terms change multiplicatively. Linear covariate rescaling automatically leads to multiplication of $\mathbf{Z}^i$ by a constant for any marginal basis itself satisfying conditions SI1 and SI2. However for most bases the $\mathbf{X}^i$ may require an arbitrary rescaling of some columns, to force covariate rescaling to induce only multiplicative changes in $\mathbf{X}^i$: this would violate the requirement that our measurements of smoothness have explicit dependence on covariate scale, required for SI1 and SI2 to be sufficient for scaling invariance. To avoid such a violation, and restore full scaling invariance we can split each margin into $\mathbf{Z}^i$ and each separate column of $\mathbf{X}^i$, as in the method variant in parentheses in step 3 and 5 above.

Note that our construction does not require that marginal P-spline penalties are actually re-written in scale dependent form in order to compute scaling invariant estimates: it suffices that SI1 and SI2 *would* be met *if* this were done.

3.2 Explicit form of the penalties

Our general construction does not make it clear what aspects of function shape are being penalized by the resulting penalties, but in fact the penalties have clear meanings in terms of function shape, if the penalties on the marginals have clear meanings. As an example, consider the widely applicable case in which the marginals have penalties based on integrals of squared derivatives. There are two cases to consider. The penalty on the product of the penalty range spaces of two marginals, and the penalty on the product of a range space and a null space.

In particular, consider marginal smooths $h(\mathbf{x})$ and $g(\mathbf{z})$, with penalties

$$\int \sum_i (D_i h)^2 d\mathbf{x} \quad \text{and} \quad \int \sum_j (\Delta_j g)^2 d\mathbf{z}$$

where the D_i and Δ_i are differential operators.

1. Let δ_k denote the k^{th} element of the operator given by $\sum_i D_i \sum_j \Delta_j$. The penalty on the product of the range spaces of the marginal smooths is

$$\int \sum_k (\delta_k f)^2 d\mathbf{x} d\mathbf{z}.$$

2. The product of the null space basis of g's penalty with the basis for h can be written

$$\sum_j \gamma_j(\mathbf{x})\tilde{c}_j(\mathbf{z})$$

where the $\tilde{c}_j(\mathbf{z})$ are basis functions for the null space of the penalty on g and $\gamma_j(\mathbf{x})$ is a smooth 'coefficient function' represented using the basis for h. The penalty on the product of the null space of g with the range space of h is then

$$\sum_j \int \sum_i (D_i\gamma_j)^2 d\mathbf{x}.$$

Construction of penalties for products of more than two marginals simply applies these rules iteratively.

Here are some examples of penalties on range space products for various marginal penalties, where a subscript variable denotes partial differentiation w.r.t. that variable:

1. For cubic spline marginal penalties $\int h_{xx}^2 dx$ and $\int g_{zz}^2 dz$ the product penalty is

$$\int f_{xxzz}^2 dxdz,$$

as Appendix 2 shows in detail. Notice how this meets SI2 under transformations of the form $x \leftarrow ax$, $z \leftarrow bz$, where a and b are positive constants.

2. For a thin plate spline with penalty $\int\int h_{xx}^2 + 2h_{xz}^2 + h_{zz}^2 dxdz$ and a cubic spline of t we have product penalty

$$\int f_{xxtt}^2 + 2f_{xztt}^2 + f_{zztt}^2 dxdzdt.$$

Notice how this meets SI2 under transformations $x \leftarrow ax$, $z \leftarrow az$ and $t \leftarrow bt$.

3. For a product of thin plate splines of x, z and v, w the product penalty is

$$\int f_{xxvv}^2 + 2f_{xxvw}^2 + f_{xxww}^2 + 2f_{xzvv}^2 + 4f_{xzvw}^2 + 2f_{xxww}^2 + f_{zzvv}^2 + 2f_{zzvw}^2 + f_{zzww}^2 dxdzdvdw.$$

This meets SI2 under transformations $x \leftarrow ax$, $z \leftarrow az$, $v \leftarrow bv$ and $w \leftarrow bw$.

(In all cases invariance obviously also holds under arbitrary translation.)

See Wahba (1990, section 10.2) and Gu (2002, Section 2.4 in particular tables 2.3 and 2.4) for more on induced penalties for this sort of construction, and Appendix 1 for some further examples.

3.2.1 Low rank SS-ANOVA

An interesting case arises from the marginal penalties $\int h_x^2 dx$. Consider creating a smooth $f(x, z)$ from such marginals: the term will be subject to 3 penalties: $\int f_x^2 dx$ penalizing model matrix columns that constitute a basis for the range space of the marginal smooth of x, $\int f_z^2 dz$ penalizing model matrix columns that constitute a basis for the range space of the marginal smooth of z, and $\int f_{xz}^2 dxdz$ penalizing columns that constitute a basis for the range space of the interaction. Hence if the smoothing parameter for the interaction penalty $\to \infty$, the smooth tends to the additive model $f_x(x) + f_z(z)$. It can readily be verified that this ANOVA decomposition approach generalizes to any number of marginals.

The $\int h_x^2 dx$ penalties result in this simple interpretation because their null space is the space of constant functions. Hence there are no "mixed" tensor products between marginal null spaces and marginal range spaces to consider in this case: the tensor products involving the marginal null spaces reduce to the marginal smooths. When other penalties are employed the natural generalization of an ANOVA decomposition to the functional setting is not quite so obvious. For example, when using cubic spline marginals, and the full construction, with penalties $\int h_{xx}^2 dx$, then $f(x, z)$ decomposes into the following separately penalized components:

$$f_x(x) + f_z(z) + f_{x1}(x)z + f_{z1}(z)x + f_{xz}(x, z).$$

We can deem that the smooth interaction of x, z is given by $f_{x1}(x)z + f_{z1}(z)x + f_{xz}(x, z)$, although the corresponding penalty is perhaps not so natural now. See Lee and Durbán (2011) and Belitz and Lang (2008) and Wood (2006a) for alternative approaches to penalizing the interaction space, and Gu (2002) for a fuller treatment of the whole subject.

3.2.2 Omitting higher order components

When smoothing with respect to many variables the number of smoothing parameters to estimate can become rather high (especially if the fully invariant version of the smooth is used with 2nd or higher order penalties), and it may be desirable to employ simplified smooths, which simply omit components of the smooth subject to high order penalties. By a component we mean a single penalty and its associated model matrix columns. The resulting smooths automatically satisfy invariance.

For example when smoothing with respect to 3 variables, we might choose to omit all components of the smooth which depend on all 3 variables, retaining only those dependent on 2 variables or fewer. For a fully invariant tensor product smooth, with second order marginal penalties, this would reduce the number of smoothing parameters to estimate from 19 to 12. Alternatively, without deleting components, the

less invariant version of the construction (which treats each null space basis 'as a whole') would require 7 smoothing parameters, as would a smooth based on first order penalties (for which both constructions coincide, and are fully scaling invariant).

3.3 Implementation in R and SAS

To illustrate the simplicity of using the new construction with existing mixed model software, we give the essential details for R packages `nlme` (Pinheiro and Bates, 2000) and `lme4` (Bates and Maechler, 2010) and for SAS. With `nlme` it is straightforward to use the `pdIdent` class to incorporate a term in a model which has i.i.d. random coefficients: all that is needed is to provide the corresponding model matrix. So the unpenalized part of the tensor product smooth gets added to the model fixed effects, while `pdIdent` terms are appended for each separately penalized submatrix of the model matrix. For detailed code see function `gamm` in R package `mgcv` from `cran.r-project.org`.

`lme4` allows separation of the model setup and model estimation phases. To incorporate a user specified general model matrix $\mathbf{Z}$ with i.i.d. coefficients, requires that we call the setup phase with an i.i.d. dummy random factor in place of the term actually required (the dummy factor should have as many levels as $\mathbf{Z}$ has columns). The object returned by the setup phase can then have the model matrix for the dummy factor variable replaced by $\mathbf{Z}$, before the fitting phase is called. We can incorporate any number of such terms in a model. For detailed code see R package `gamm4` from `cran.r-project.org`.

In SAS procedures MIXED and GLIMMIX then the columns of fixed effects model matrix, $\mathbf{X}$, are supplied in the `model` statement, while the columns of random effect model matrices, $\mathbf{Z}$, are supplied to the `random` statement. The i.i.d. structure for the corresponding random coefficients is supplied via the `type=toep(1)` option. See Appendix B.3.3 of Ruppert, Wand and Carroll (2003) for code that can be adapted for this purpose.

Fully automated section 3 construction is provided by `t2` model terms in the various additive modelling routines provided in R packages `mgcv` and `gamm4`. Actual smoother construction is handled by the function `smooth.construct.t2.smooth.spec`. Any mix of singly penalized smoothing basis available in `mgcv` can be used as a marginal bases. Note that for large datasets, `t2` terms resulting in many penalties may become infeasibly memory intensive when used with the `mgcv gam` function (`bam` may alleviate this). This is because of linear dependence of the `gam` method memory requirements on the number of smoothing parameters: other mixed model software will generally be more memory efficient in this context.

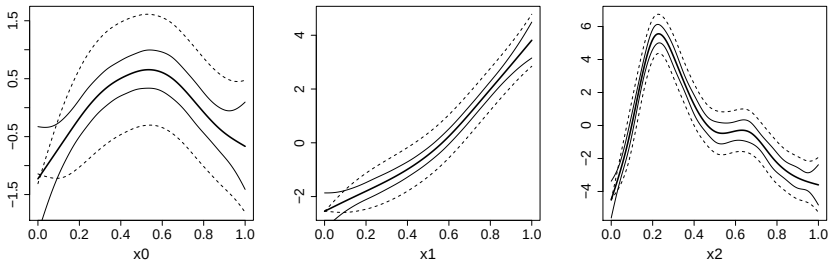

Figure 2: Comparison of component wise confidence intervals, under alternative identifiability constraints. GAMs were fitted to noisy data from a 3 term additive truth, with smoothness selection by REML. The thick curves show the indistinguishable (centred) function estimates under the alternative constraints. The thin continuous curves delimit the 95% Bayesian confidence intervals under the identifiability constraints $\sum_i f_j(x_{ji}) = 0$, while the dashed curves are the equivalent under identifiability constraints $f_j(x_{j1}) = 0$.

4 Identifiability constraints

Usually the f_j in (1) are not identifiable without constraints. For example in the model

$$y_i = \alpha + f_1(x_i) + f_2(z_i) + \epsilon_i,$$

f_1 and f_2 are confounded with the intercept, α, and therefore require identifiability constraints.

Given smoothing parameters, all alternative identifiability constraints yield the same $\hat{f}_j$ up to additive constants, and exactly the same fitted values. However, confidence intervals for the f_j are different in the space of different constraints, and poor constraint choice can lead to practically useless intervals, as figure 2 demonstrates. Since excessively wide intervals arise through linear dependence on the intercept, it is preferable to force orthogonality to the intercept via $\mathbf{1}^{\mathrm{T}}\mathbf{f} = 0$, which is the sum-to-zero constraint $\sum_i f(x_i) = 0$ (for the simple example given above such constraints would imply $\hat{\alpha} = \bar{y}$, but this need not hold in general).

Unfortunately, constraints that are equivalent in yielding identical fitted values, need not always yield identical smoothing parameter estimates. Prediction error smoothing parameter estimation criteria, such as GCV and AIC, that essentially depend only on the fitted values, yield smoothing parameter estimates that are invariant to the identifiability constraint used. Appendix 3 shows that REML also has this property, but maximum likelihood estimation of smoothing parameters is not invariant. This is because in the mixed model setting only the fixed effects component of a smooth actually suffers from an identifiability

problem. The random effects are always identifiable. However a sum to zero constraint involves both fixed and random components of a smooth, and is hence more than an identifiability constraint in the mixed setting: in fact it changes the likelihood. To see this consider

$$y_i = \alpha + f(x_i) + \epsilon_i$$

where f is a penalized regression spline. In mixed model form, this is

$$\mathbf{y} = \mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\epsilon}.$$

$\mathbf{X}$ and $\mathbf{1}$ will not be identifiable, so a constraint is needed, but imposing $\mathbf{1}^{\mathrm{T}}\mathbf{f} = 0$ will result in $\mathbf{Z}$ being replaced by its column centered version, *with no change in the distribution of* $\mathbf{b}$. So the modeled distribution of $\mathbf{y}$ is changed along with the maximum likelihood estimates of smoothing parameters, as is easily verified by example. To avoid such an arbitrary change in likelihood, constraints should be applied only to the fixed effects.

This lack of invariance poses a practical problem because R package `lme4` uses maximum likelihood estimation in the generalized case with no REML option, while SAS proc GLMMIX defaults to ML, and it is unclear whether its 'REML-like' approximation is invariant. There are two options for progressing in practice.

1. Impose a sum to zero constraint on each smooth by imposing a sum to zero constraint on the fixed effects (see e.g. Wood, 2006b, section 4.2) and subtracting the column mean from each column of the random effect model matrices. This approach accepts that the constraint is more than is needed for identifiability, but uses it in order to obtain interpretable intervals for the smooth components of the model.

2. Perform estimation with identifiability constraints on the fixed effects alone. Subsequently re-parameterize to obtain the model coefficient estimates that would have been obtained under sum to zero constraints. This approach avoids arbitrary modification of the likelihood used for smoothing parameter estimation, while still providing interpretable intervals for the smooth components.

Option 1 is simpler to implement, and in our experience produces reasonable results, not substantially different from option 2. However simulation testing also suggests that option 2 produces lower mean square error than option 1. The details for option 2 are provided in appendix 4, and it is used for the rest of this paper.

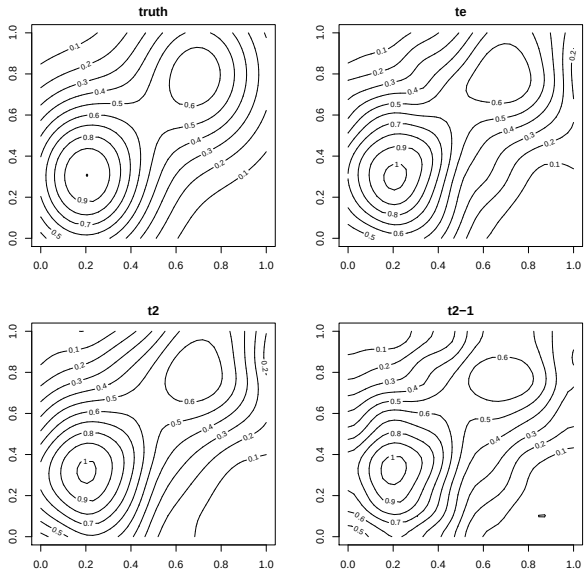

Figure 3: Reconstructing a non additive truth by several tensor product smoothing methods. Top left is the truth. The remaining panels are reconstructions from 400 noisy samples ($\sigma = 0.1$), using different tensor product smoothers. Top right is a Wood (2006a) type smooth, with cubic regression spline marginals. Bottom left is the equivalent constructed by the section 3 method (the two variants give indistinguishable results in this case). Bottom right is the section 3 method using first order thin plate spline marginals, as in section 3.2.1. See section 5 for more detail.

5 Simulation comparisons

The objective of this paper is to produce tensor product smooths that can be used with modern mixed modeling software, not to produce smooths that have better or worse statistical performance than existing methods. We therefore provide only limited simulation comparison with the method given in Wood (2006a), which in turn provides comparison with other alternatives.

Simulations were conducted as follows. For each replicate, 400 values of covariates x and z were simulated from independent $U(0, 1)$ distributions. Two 'true' functions were evaluated at the simulated x, z values. The non-additive truth shown at the top left of figure 3 was one, and the other was the additive truth:

$$0.2x^{11}\{10(1-x)\}^6 + 10^4x^3(1-x)^{10} + \exp(2z)$$

rescaled to have a range of 0 to 1. The response data were generated by adding i.i.d gaussian noise to the sampled truth, at each of levels $\sigma = 0.05$, 0.1, 0.2 and 0.5. Only one sample size is reported, since increasing or decreasing the sample size gives results almost indistinguishable from decreasing or

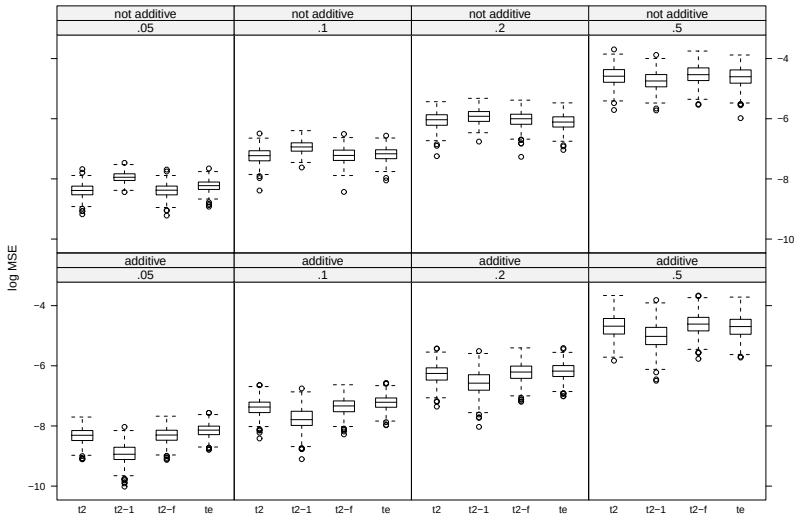

Figure 4: Summary of the distribution of log MSE from the simulation described in section 5 (where the labels are explained). Clearly the section 3.2.1, 't2-1' smooths are able to pick out an additive truth very well, but do less well when the truth is not additive.

increasing the noise level by some amount.

For each underlying truth, at each noise level, four alternative tensor product smoothers were fitted to the simulated data:

'te' used cubic regression spline marginals and the Wood (2006a) construction.

't2' used cubic regression spline marginals and the section 3 construction.

't2-f' used cubic regression spline marginals and the fully invariant alternative construction of section 3.

't2-1' used thin plate spline marginals with a first derivative penalty and the construction of section 3.

10 dimensional marginals were used in each case. 500 replicates of each combination of truth and noise level were run, with all 4 smooths fitted to each, using REML smoothness selection. Figure 3 shows a typical set of reconstructions of the non-additive truth, for $\sigma = 0.1$.

The results shown in figures 4 and 5 suggest that the ANOVA decomposition character of the section 3 smooths conveys an advantage when the truth is additive, with this being particularly marked for the most ANOVA like smooth 't2-1'. The improvement of 't2-1' over 't2' appears to relate to the fact that 't2' can not penalize away terms of the form $f(x)z$ and $f(z)x$ without also penalizing away the required terms $f(x)$ and $f(z)$. The improvement of 't2-1' over 't2-f' appears to be caused by 't2-f' having 3

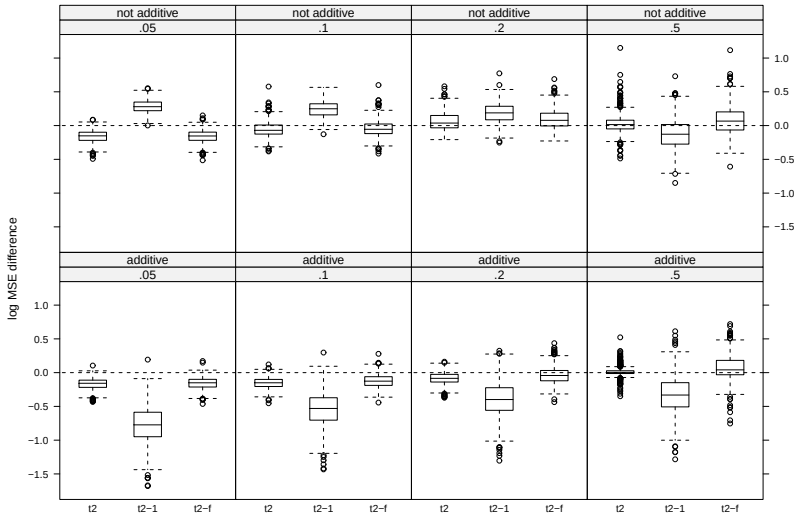

Figure 5: log MSE of the three section 3 smooths with the corresponding log MSE of the Wood (2006a) smooth subtracted. Lower is better, and negative means that the new smooth was better than Wood (2006a). The results emphasize that the section 3.2.1 't2-1' smooth has an advantage when the truth is additive, but all the section 3 smooths show some advantage in this case, reflecting their 'ANOVA-decomposition' structure. The non-additive case is less clear cut. It seems that in high information situations the new smooths 't2' and 't2-f' give better results, but at high noise the advantage is reversed.

smoothing parameters associated with the smooth interaction of x and z while 't2-1' has only 1. In consequence 't2-1' terms are more often estimated to have no smooth interaction than 't2-f', when the truth is additive. A further slight advantage is conferred by the boundary behaviour of the 't2-1' penalties in the additive truth case: the penalty favours functions that tend to constants at the boundaries, and this is advantageous for 3 of the 4 boundaries in this particular simulated additive case.

However 't2-1' is the worst performer in non-additive situations, apparently because of poor boundary performance, resulting from the use of first order derivative penalties penalizing towards a constant at the boundaries (see figure 3 bottom right). For the non-additive case the 't2' and 't2-f' smooths give similar results, and improve on the Wood (2006a) 'te' terms at low noise. At high noise the advantage is reversed, with the 'te' smooth giving the better performance. It is possible that at low noise the extra model flexibility engendered by having more penalties is an advantage, whereas at high noise it is disadvantageous to have to estimate relatively many smoothing parameters. Recall that the 'te' smooth has 2 penalties, 't2' has 3 and 't2-f' has 5. We suspect that the good performance of the 't2-1' smooth on the non-additive truth at the highest noise level is an anomaly of the particular true function.

6 Examples

This section presents two brief examples of the type that originally motivated this work, highlighting the practical advantages that follow from being able to use tensor product smooths with a wider range of mixed model software than had been possible hitherto. The first example illustrates the improved MSE performance and improved estimation reliability achievable in the context of binary data, as a result of being able to avoid PQL. The second example shows how use of the method allows access to better methods for modelling over-dispersion, and improved model comparison via access to AIC values. For discussion of the advantages that might result from the functional ANOVA interpretation of the smooths, we refer to Gu (2002).

6.1 A simulated presence-absence survey

The invariant tensor product smooths in Wood (2006a) can only be estimated as mixed model components using the `nlme` package in R. This means that their use as generalized additive mixed model components requires the use of PQL (Breslow and Clayton, 1993) for estimation. This is known to be problematic for binary data. The new construction allows estimation using the more modern methods available in `lme4`. To illustrate the advantage that this represents we present a simulation based on surveying for the

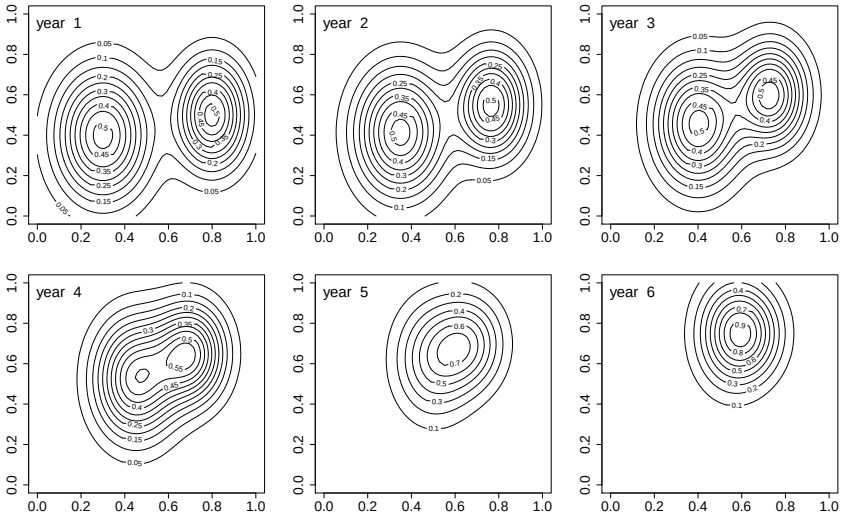

Figure 6: True probability of presence through time for the simulated survey example in section 6.1.

presence or absence of a species within some area. The basic setup is that every year a survey is made in which 200 quadrats are randomly selected from a total of 1600 quadrats in the area, to be visited by volunteers, who record the presence or absence of the species of interest. There are 200 volunteers, but only half are selected each year, and each visits two quadrats. There are 6 years of data in total (i.e. 1200 observations). There are random observer effects (which may be positive or negative – observers may miss the species if present, and may mis-identify and record a presence in place of an absence). The true probability of presence per quadrat is shown in figure 6 for each year. The idea is that the species range is contracting and shifting north. To get the probability that an observer detects the species, we take the logit of the plotted probability, add the observer effect and apply the logistic function (inverse of the logit) to the result. The observer effects are i.i.d. $N(0, .2^2)$.

Presence/absence data, y_i, were simulated under this setup, and fit by the following model:

$$\text{logit}(\mu_i) = f(x_i, z_i, t_i) + b_{k(i)}$$

where the y_i are independent Bernoulli random variables with expected value μ_i. x_i, z_i is the quadrat centre of the i^{th} measurement, t_i is time and $k(i)$ is the index of the observer who made the i^{th} observation. The b_k are i.i.d. $N(0, \sigma_b^2)$ observer random effects. Smooth function f is represented by a section 3 tensor product smooth based on a rank 30 thin plate regression spline of x_i, z_i and a rank 5 cubic regression spline of t_i. Note that in `mgcv`/`gamm4` such a smooth is incorporated in a model formula with a term like `t2(x,z,t,d=c(2,1),k=c(30,5),bs=c("tp","cr"))` (`d` specifies the number of

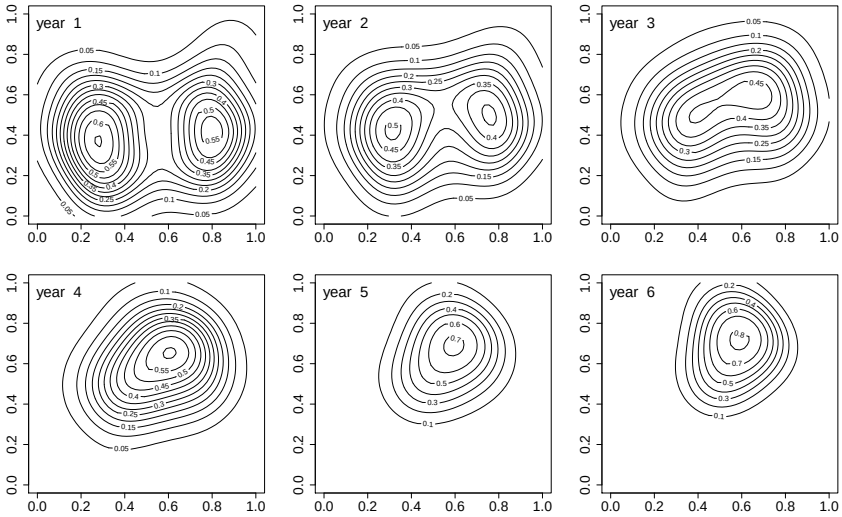

Figure 7: Reconstructed probability of presence through time (figure 6) from a typical `lme4` model fit from section 6.1.

covariates for each margin, `k` the corresponding marginal basis dimension and `bs` the type of basis). The model was estimated in two ways: by PQL as implemented in `mgcv` function `gamm` and by the full Laplace approximation based method implemented in `lme4`. The mean square error between the true μ_i and the fitted $\hat{\mu}_i$ was then assessed for each fitting method (if it converged successfully).

50 replicates of the simulation were run. Figure 7 shows a typical reconstruction using `lme4` based fitting. 12 PQL fits failed. Across the remaining replicates the average MSE for the `lme4` fit was 48% of the PQL equivalent (range 21% to 85%). In short the fact that the new construction enables us to fit the model using improved modern mixed modeling methods leads to substantially improved performance both statistically and computationally.

6.2 Mackerel eggs

The second example uses real over-dispersed data from a survey of mackerel eggs. Although there are several ways of dealing with overdispersed count data, an appealing approach is to allow a random effect per count (see e.g. Davison, 2003, Section 10.6). This example illustrates a case where this is possible, given the new construction, where it was not before. The data are from a survey of Mackerel eggs conducted off the west coast of Britain and Ireland in 1992, the response (`egg.count`) being number of eggs found in survey nets hauled through the water at various sampling stations (see Borchers et al. 1997 for more information). They are modeled in Wood (2008) using an additive model consisting of a

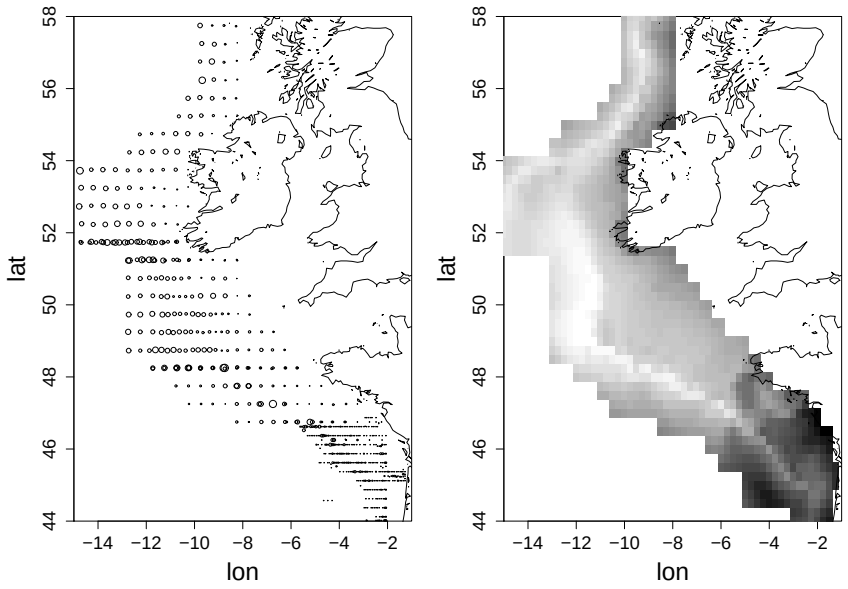

Figure 8: Left panel: Mackerel egg survey data modeled in section 6.2. Symbol sizes are proportional to square root of egg count. Right panel: log density of eggs per m^2 of sea surface according to the model presented in section 6.2.

spatial smoother and smooths of water temperature at 20m (`temp.20m`) and sea bed depth (`b.depth`). Apparent overdispersion in the data was accommodated by a quasi-Poisson approach.

This model was somewhat unsatisfactory for two reasons. Firstly the use of a quasi-likelihood approach complicates model selection, since AIC based model comparison is precluded. Secondly, it would be preferable for the model to be based more closely on the covariates that biologists believe the fish are responding to, rather than relying on spatial location. This latter point is particularly significant in this context since the covariates are heavily confounded with spatial location, which undermines the stability of the model estimates.

A model that is more biologically based uses the fact that Mackerel are known to favor the continental shelf edge, so that a tensor product smooth of distance (`c.dist`) from the 200m depth contour and latitude (`lat`) might result in a more reliable and interpretable model than a simple spatial smooth (the 200m depth contour is a good proxy for the shelf edge). So the following model is proposed

$$\log(\mu_i) = f_1(\texttt{c.dist}_i, \texttt{lat}_i) + f_2(\texttt{temp.20m}_i) + f_3(\sqrt{\texttt{b.depth}_i}) + \log(\texttt{net.area}_i) + e_i \qquad (3)$$

where the e_i are i.i.d. $N(0, \sigma_e^2)$ random effects used to model overdispersion, and $\texttt{egg.count}_i$ are independent Poi(μ_i). The f_j are smooth functions, with f_1 represented using a section 3 tensor product smooth based on marginal rank 10 cubic regression splines. The offset term allows for the fact that

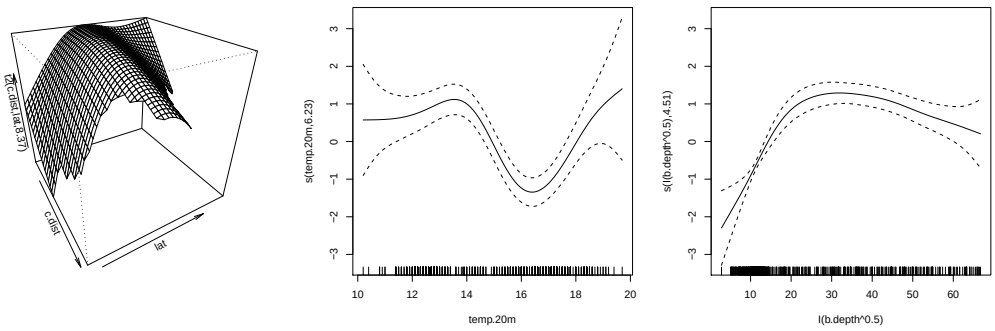

Figure 9: Estimated smooth effects from the model reported in section 6.2. The left panel is a tensor product constructed using the method of section 3 (only plotted in the vicinity of supporting data). The model was estimated using `lme4`.

different net diameters were used at different sample stations.

Note that we were unable to fit (3) by PQL (the working model estimate failed at the third iteration), but fitting by `lme4` was unproblematic, and results in the effect estimates shown in figure 9 and the density predictions plotted on the right panel of figure 8.

This approach immediately enables us to formally check for over-dispersion by comparing AIC values for the model with and without the observation specific random effect. The AIC difference is extreme (> 10000), confirming the overdispersion strongly suggested from residual plots.

A more substantial benefit of the approach is that it allows straightforward comparison of model (3) with a strictly additive version. The estimates of (3) suggest that most of the degrees of freedom for the tensor product smooth are in the interaction components of the smooth, but it is still of interest to compare with the more parsimonious model in which the smooth effects of `c.dist` and `lat` are strictly additive (the additive version requires 5 fewer smoothing parameters than model (3)). In fact the AIC values for the competing models differ by less than 0.5, suggesting that either could be used: a conclusion that could not have been reached if we had had to rely on PQL.

7 Discussion

The methods proposed here meet the paper's aims of producing a tensor product construction that gives rise to smoothers that can readily be estimated using modern standard mixed modeling software, while maintaining the key properties of scale invariance and interpretable smoothing penalties. As the examples

show, this has the practical advantage of allowing the best mixed model estimation methods to be used for models involving these terms, which can improve both statistical performance and computational reliability.

The proposed method follows most closely from the smoothing spline ANOVA approach as described in Gu (2002) or Wahba (1990, section 10.2), which also decomposes functions into separate singly penalized subspaces, constructed from the product of penalty null space and range space penalties, and results in penalties with similar interpretations to ours. Like Gu 2002 (or Lee and Durbán, 2011 or Belitz and Lang 2008, or Wood 2006a) our construction also allows ANOVA type decompositions of function estimates, but previous approaches did not allow estimation with standard mixed model software: previous tensor product smooths based on reduced rank spline marginals did not allow separation of penalties, while Kim and Gu's (2004) reduced rank approach to SS-ANOVA involves the smoothing parameters entering both the basis and the penalty. In contrast, our construction can readily be incorporated into mixed models to be estimated by a wide range of standard software, incorporating the most modern estimation methods.

Our primary aim was to produce scaling invariant smooths that can be estimated by standard mixed modelling methods, rather than smooths that have better or worse performance than other scaling invariant smooths, but, as the simulation results emphasise, the smooths presented here will outperform alternatives in some circumstances. The most obvious is when the truth is in a form exactly representable by the model, such as additive, or additive plus varying coefficient interactions (e.g $f_1(x) + f_2(z) + zf_3(x) + xf_4(z)$). Of course it as an open question how often nature arranges matters so conveniently? The second possible advantage arises when the most appropriate penalty is not known, so that the flexibility engendered by a penalty with several smoothing parameters offers the potential to better approximate the truth. When the data signal to noise ratio is high, so that the smoothing parameters can be well estimated, this may confer a practical advantage, although at low signal to noise ratios the extra variability in estimating the smoothing parameters is likely to reverse this.

The methods reported here are available as the `t2` smooth class in R packages `mgcv` and `gamm4`, available from `cran.r-project.org` (R Development Core Team, 2010).

Acknowledgements

We thank Maria Durbán for an early pre-print of Lee and Durbán (2011), and some discussion of P-spline ANOVA. We are very grateful to two anonymous referees for many useful comments and suggestions, in particular on scaling invariance and the possibility of dropping high order components of smooths.

Appendix 0: conditions for scaling invariance

This appendix demonstrates that SI1 and SI2 in section 2.3 are sufficient to achieve scaling invariance. Recall that we assume, without loss of generality, that all smooths are written in a form where we make explicit the dependence on covariate scale of their measurement of function variability with respect to a covariate. For P-spline penalties this simply requires that we include dependence on knot spacing in the difference penalties on the coefficients. e.g. $\sum_i (\beta_{i+1} - \beta_i)^2$ would become $\sum_i (\beta_{i+1} - \beta_i)^2/h^2$ where h is the P-spline knot spacing (division by h rather than h^2 can also be justified). This assumption makes no difference to the performance of the marginal smooths as one dimensional smoothers, but in the multi-dimensional case makes explicit the relative weighting given to smoothness with respect to different variables. Note that re-parameterization as in section 2.5 will transfer the dependence on covariate scale from the penalty matrix to the basis: all that matters from our point of view is that the dependence remains.

Generally we are interested in the case in which the smooth is part of a larger model, containing parametric effects and other smooths. Possibly after re-parameterization the linear predictor of such a model can be written

$$\boldsymbol{\eta} = \mathbf{A}\boldsymbol{\theta} + \sum_j \mathbf{Z}_j \mathbf{b}_j$$

where $\mathbf{A}$ and the $\mathbf{Z}_j$ are model matrices, which may be combined into a single model matrix $\mathbf{X} = [\mathbf{A} : \mathbf{Z}_1 : \cdots]$ and $\boldsymbol{\theta}$ and the $\mathbf{b}_j$ are coefficient vectors, which may be stacked into one vector $\boldsymbol{\beta}$. The penalty associated with the model can be written as

$$\boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_\lambda \boldsymbol{\beta} = \sum_j \sum_k \lambda_{jk} \mathbf{b}_j^{\mathrm{T}} \mathbf{S}_{jk} \mathbf{b}_j.$$

Again without loss of generality we assume that any fitting weights have been absorbed into $\mathbf{X}$ via $\mathbf{X} \leftarrow \sqrt{\mathbf{W}}\mathbf{X}$.

Given scale explicit penalties, and an identifiable model, scaling invariance will occur if our inferences about $\boldsymbol{\eta}$ are invariant to linear rescaling of the smoothing covariates. For this to happen it is sufficient that the influence matrix (hat matrix) for the model is unchanged by covariate rescaling. We now demonstrate that this is the case. In the following let primed quantities denote versions under transformation.

Under SI1 from section 2.3, we have that $\mathbf{X}' = \mathbf{X}\mathbf{C}$ where $\mathbf{C}$ is a diagonal matrix of coefficients. The subset of elements of $\mathbf{C}$ corresponding to any one $\mathbf{b}_j$ all have the same value. So the transformed influence matrix is given by

$$\begin{aligned}
\mathbf{A}' &= \mathbf{X}'(\mathbf{X}'^T\mathbf{X}' + \mathbf{S}'_\lambda)^{-1}\mathbf{X}'^T = \mathbf{X}\mathbf{C}(\mathbf{C}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{C} + \mathbf{S}'_\lambda)^{-1}\mathbf{C}\mathbf{X}^{\mathrm{T}} \\
&= \mathbf{X}\mathbf{C}\mathbf{C}^{-1}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \mathbf{C}^{-1}\mathbf{S}'_\lambda\mathbf{C}^{-1})^{-1}\mathbf{C}^{-1}\mathbf{C}\mathbf{X}^{\mathrm{T}} \\
&= \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{X}^{\mathrm{T}} = \mathbf{A}
\end{aligned}$$

if

$$\mathbf{S}_\lambda = \mathbf{C}^{-1}\mathbf{S}'_\lambda\mathbf{C}^{-1}. \tag{4}$$

It turns out that (4) can always be made to hold via appropriate choice of the smoothing parameters λ_{jk}. First note that $\mathbf{S}_\lambda$ is block diagonal with blocks of the form $\sum_k \lambda_{jk}\mathbf{S}_{jk}$. By SI1 the diagonal elements of $\mathbf{C}$ corresponding to

a block are a constant, c_j, say. By SI2 we have that $\mathbf{S}'_{jk} = \mathbf{S}_{jk}d_{jk}$ where d_{jk} is a constant. So $\mathbf{C}^{-1}\mathbf{S}'_\lambda\mathbf{C}^{-1}$ is made up of blocks of the form $\sum_k \lambda'_{jk}\mathbf{S}_{jk}d_{jk}/c_j^2$. These blocks are equal to $\sum_k \lambda_{jk}\mathbf{S}_{jk}$ if $\lambda'_{jk} = \lambda_j c_j^2/d_{jk}$, which is always possible since the smoothing parameters are unconstrained. Hence (4) can always hold, and inference will be scaling invariant.

Appendix 1: Explicit construction example

In order to achieve generality, the section 3 presentation of the new construction is somewhat abstract. This appendix provides an explicit example for a two dimensional smooth, $f(v, w)$, based on spline marginals. Here we will generally denote smooth functions by f under the understanding that functions differ if their arguments differ. We will denote row-wise Kronecker products by $\otimes_r$, so $\mathbf{A} \otimes_r \mathbf{B}$ denotes the matrix whose i^{th} row is the Kronecker product of the i^{th} rows of $\mathbf{A}$ and $\mathbf{B}$.

Let the first marginal smooth be $f(v)$, with marginal model matrix $\boldsymbol{\mathcal{X}}_v$ and penalty $\mathcal{P}_v = \int f_{vv}^2 dv$ (where f_{vv} denotes the second derivative of f with respect to v). If $\boldsymbol{\beta}_v$ are the coefficients of $f(v)$ then $\mathcal{P}_v = \boldsymbol{\beta}_v^{\mathrm{T}}\mathbf{S}_v\boldsymbol{\beta}_v$, where $\mathbf{S}_v$ is a matrix of fixed coefficients.

Similarly the second marginal smooth is $f(w)$, with marginal model matrix $\boldsymbol{\mathcal{X}}_w$ and penalty $\mathcal{P}_w = \int f_w^2 dw = \boldsymbol{\beta}_w^{\mathrm{T}}\mathbf{S}_w\boldsymbol{\beta}_w$: to make the example more instructive, this is a first order penalty.

The first step in the construction is to reparameterize the marginal bases, using the symmetric eigen-decomposition (identical to the SVD here, but more computationally efficient) of the penalties

$$\mathbf{S}_\cdot = \mathbf{U}_\cdot\boldsymbol{\Lambda}_\cdot\mathbf{U}_\cdot^{\mathrm{T}}$$

where $\mathbf{U}_\cdot$ is the orthogonal matrix of eigenvectors and $\boldsymbol{\Lambda}_\cdot$ the diagonal matrix of corresponding eigenvalues. Now $\bar{\boldsymbol{\Lambda}}$ is the diagonal matrix such that $\bar{\Lambda}_{ii} = \sqrt{\Lambda_{ii}}$ if $\Lambda_{ii} > 0$ and $\bar{\Lambda}_{ii} = 1$ if $\Lambda_{ii} = 0$. We then reparameterize so that the model matrices become $\boldsymbol{\mathcal{X}}'_\cdot = \boldsymbol{\mathcal{X}}_\cdot\mathbf{U}_\cdot\bar{\boldsymbol{\Lambda}}_\cdot^{-1}$.

Under the re-parametization $\mathbf{S}_v$ becomes the identity matrix, with the last two elements on the leading diagonal set to zero, while $\mathbf{S}_w$ becomes the identity matrix with the last element on the leading diagonal set to 0. Of course the penalties are unchanged by this in that they still measure the same things about the marginal smooths.

Now the marginal model matrices are partitioned into penalized columns $\mathbf{Z}_\cdot$ and unpenalized (null space basis) columns $\mathbf{X}_\cdot$:

$$\boldsymbol{\mathcal{X}}'_\cdot = [\mathbf{Z}_\cdot : \mathbf{X}_\cdot].$$

Here $\mathbf{X}_w = \mathbf{1}$ while $\mathbf{X}_v = [\mathbf{v} : \mathbf{1}]$, where $\mathbf{1}$ is a column of 1s and $\mathbf{v}$ is a column containing the observed v values (we have here assumed that the null space re-parameterization of section 2.5 has been applied to the null space of $\mathcal{P}_v$, and without loss of generality have ignored any multiplicative factors on the null space columns).

The first variant of our construction would then result in a tensor product model matrix that can be partitioned as follows

$$\mathbf{X} = [\mathbf{Z}_v \otimes_r \mathbf{Z}_w : \mathbf{Z}_v \otimes_r \mathbf{X}_w : \mathbf{Z}_w \otimes_r \mathbf{X}_v : \mathbf{X}_w \otimes_r \mathbf{X}_v]$$

Each of the three left most partitions can now be treated as relating to a block of i.i.d. Gaussian random effects, while the coefficients for the final block are treated as fixed effects. The penalized blocks form bases for different spaces and have different penalties as listed below (working through the blocks from the left):

1. Basis for functions of v and w (excluding basis components from later blocks), penalized by $\int f_{vvw}^2 dvdw$.
2. Basis for functions of v, penalized by $\int f_{vv}^2 dv$.
3. Basis for functions of the form $f(w) + g(w)v$, penalized by $\int f_w^2 dw + \int g_v^2 dv$.

All bases listed exclude null space components, which are collected in the final unpenalized block.

The final penalty above makes it clear that this construction is not fully scaling-invariant. The fully invariant construction (now giving columns of $\mathbf{X}$. explicitly) gives

$$\mathbf{X} = [\mathbf{Z}_v \otimes_r \mathbf{Z}_w : \mathbf{Z}_v : \mathbf{Z}_w \otimes_r \mathbf{v} : \mathbf{Z}_w : \mathbf{X}_w \otimes_r \mathbf{X}_v]$$

The four left most partitions can now be treated as relating to a block of i.i.d. Gaussian random effects. The block interpretations and penalties are now :

1. Basis for functions of v and w (excluding basis components from later blocks), penalized by $\int f_{vvw}^2 dvdw$.
2. Basis for functions of v, penalized by$\int f_{vv}^2 dv$.
3. Basis for the space of functions of the form $f(w)v$, penalized by $\int f_w^2 dw$.
4. Basis for functions of the form $f(w)$, penalized by $\int f_w^2 dw$.

Again these bases exclude null space components.

Appendix 2: A detailed penalty example

To better understand the product penalty, consider the case of two marginal cubic spline penalties. Suppose the marginal basis expansions are $g(z) = \sum_j \gamma_j c_j(z)$ and $h(x) = \sum_i \alpha_i a_i(x)$, where γ_j and α_i are coefficients while c_j and a_i are basis functions. Assume, without loss of generality, that the range and null space of the penalty have separate basis functions, and that the penalty reduces to a ridge penalty on the range space coefficients. The expansions can be re-written in vector notation as $g(z) = \mathbf{c}(z)^{\mathrm{T}}\boldsymbol{\gamma}$ and $h(x) = \mathbf{a}(x)^{\mathrm{T}}\boldsymbol{\alpha}$, in which case the penalties become

$$\int g_{zz}^2 dz = \boldsymbol{\gamma}^{\mathrm{T}} \int \mathbf{c}_{zz}\mathbf{c}_{zz}^{\mathrm{T}} dz \boldsymbol{\gamma} = \boldsymbol{\gamma}^{\mathrm{T}} \boldsymbol{I}_z \boldsymbol{\gamma} \text{ and } \int h_{xx}^2 dx = \boldsymbol{\alpha}^{\mathrm{T}} \int \mathbf{a}_{xx}\mathbf{a}_{xx}^{\mathrm{T}} dx \boldsymbol{\alpha} = \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{I}_x \boldsymbol{\alpha},$$

where $\mathbf{c}_{zz}$ and $\mathbf{a}_{xx}$ are vectors containing $c''(z)$ and $a''(x)$, respectively. Now the tensor product basis is

$$f(x,z) = \sum_{ij} \beta_{ij} c_j(z) a_i(x) = \mathbf{b}(x,z)^{\mathrm{T}}\boldsymbol{\beta}$$

where $\mathbf{b}^{\mathrm{T}} = (a_1c_1, a_1c_2, \ldots a_2c_1, a_2c_2, \ldots)$ and $\boldsymbol{\beta}^{\mathrm{T}} = (\beta_{11}, \beta_{12}, \ldots)$. The product penalty is

$$\begin{aligned}\int\int f^2_{xxzz}dxdz &= \boldsymbol{\beta}^{\mathrm{T}}\int\int \mathbf{b}_{xxzz}\mathbf{b}^{\mathrm{T}}_{xxzz}dxdz\boldsymbol{\beta} = \boldsymbol{\beta}^{\mathrm{T}}\int\int(\mathbf{a}_{xx}\mathbf{a}^{\mathrm{T}}_{xx})\otimes(\mathbf{c}_{zz}\mathbf{c}^{\mathrm{T}}_{zz})dxdz\boldsymbol{\beta}\\ &= \boldsymbol{\beta}^{\mathrm{T}}\int\int \mathbf{I}_x\otimes\mathbf{I}_z dxdz\boldsymbol{\beta} = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{I}_{xz}\boldsymbol{\beta}\end{aligned}$$

Where $\mathbf{I}_{xz}$ is the identity matrix with leading diagonal elements zeroed, unless they penalize coefficients relating to the product of two range space basis functions, and $\otimes$ denotes the Kronecker product. In other words the ridge penalty on the range space product is $\int\int f^2_{xxzz}dxdz$.

Appendix 3: Invariance of REML to sum to zero constraints

This appendix shows that under REML smoothing parameter estimation, sum to zero constraints on model components produce identical smoothing parameter estimates to identifiability constraints applied only to the fixed effect components of smooths. Following Wood (2011), a Laplace approximate generalized restricted likelihood has the form

$$2l(\hat{\boldsymbol{\beta}}) + \log|\mathbf{S}/\phi|_+ - \hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}/\phi - \log|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}/\phi| + M_p\log(2\pi) \tag{5}$$

where $\mathbf{X}$ is the model matrix (fixed and random effect combined), $\boldsymbol{\beta}$ is the vector of fixed and random coefficients, l is the log likelihood, $\mathbf{S}$ is a the total penalty matrix so that the $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}$ is the penalty on the likelihood, ϕ is the scale parameter and $\mathbf{W}$ is a diagonal matrix, the elements of which depend on the $\mathbf{X}\boldsymbol{\beta}$. M_p is the dimension of the null space of $\mathbf{S}$. $\mathbf{S}$ is positive semi-definite, and any $\boldsymbol{\beta}$ corresponding to the constant function is in its null space. (5) is exact for a linear mixed model.

Without loss of generality assume that the first M_p columns of $\mathbf{X}$ correspond to unpenalized fixed effects. The difference between applying identifiability constraints and sum to zero constraints then reduces to the difference between column centering some penalized columns of $\mathbf{X}$, or not. $l(\hat{\boldsymbol{\beta}})$, $|\mathbf{S}/\phi|_+$ and $\hat{\boldsymbol{\beta}}^{\mathrm{T}}\mathbf{S}\hat{\boldsymbol{\beta}}$ are invariant to such a change. The following shows that $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}/\phi|$, and hence (5), are also invariant.

Theorem 1. *Let $\mathbf{X}$ be an $n \times p$ matrix assumed without loss of generality to have first column $\mathbf{1}$, $\mathbf{S}$ be a $p \times p$ positive semi-definite matrix with first row and column zero and $\mathbf{W}$ be a diagonal weight matrix. Let $\bar{\mathbf{X}}$ be $\mathbf{X}$ with some of its columns, but not the first, modified by subtraction of constants. All are finite real matrices.*

$$|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}| = |\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \mathbf{S}|.$$

Proof. Let $\mathbf{B}$ be any square root of $\mathbf{S}$ such that $\mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{S}$ and the first column of $\mathbf{B}$ is zero. $\mathbf{B}$ exists since $\mathbf{S}$ is positive semi-definite. Form the QR decomposition

$$\begin{bmatrix}\mathbf{W}\mathbf{X}\\ \mathbf{B}\end{bmatrix} = \mathbf{Q}\mathbf{R}.$$

Let $\alpha_2, \ldots, \alpha_p$ be finite real constants, some of which may be zero, so that $\bar{\mathbf{X}} = [\mathbf{1}, \mathbf{X}_{\cdot 2} - \alpha_2\mathbf{1}, \mathbf{X}_{\cdot 3} - \alpha_3\mathbf{1}, \ldots]$. Similarly define $\bar{\mathbf{R}} = [\mathbf{R}_{\cdot 1}, \mathbf{R}_{\cdot 2} - \alpha_2\mathbf{R}_{\cdot 1}, \mathbf{R}_{\cdot 3} - \alpha_3\mathbf{R}_{\cdot 1}, \ldots]$. It is immediate that

$$\begin{bmatrix} \mathbf{W}\bar{\mathbf{X}} \\ \mathbf{B} \end{bmatrix} = \mathbf{Q}\bar{\mathbf{R}}.$$

By similar construction, form

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{B} \end{bmatrix} = \mathbf{Q}'\mathbf{R}' \Rightarrow \begin{bmatrix} \bar{\mathbf{X}} \\ \mathbf{B} \end{bmatrix} = \mathbf{Q}'\bar{\mathbf{R}}'.$$

Hence $|\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X} + \mathbf{S}| = |\mathbf{R}^{\mathrm{T}}\mathbf{R}'| = |\mathbf{R}||\mathbf{R}'|$ and $|\bar{\mathbf{X}}^{\mathrm{T}}\mathbf{W}\bar{\mathbf{X}} + \mathbf{S}| = |\bar{\mathbf{R}}^{\mathrm{T}}\bar{\mathbf{R}}'| = |\bar{\mathbf{R}}||\bar{\mathbf{R}}'|$. By standard results (e.g. Harville, 1997, Thm 13.2.10) $|\bar{\mathbf{R}}| = |\mathbf{R}|$ and $|\bar{\mathbf{R}}'| = |\mathbf{R}'|$, and the result follows. □

Note the result's limitations: it holds for linear mixed models, and when the extension of REML to the generalized case results in (5). The invariance of other generalizations of REML is an open question. Note also that REML itself is well known not to be invariant to alternative constraints on the fixed effects (e.g. corner point and sum to zero constraints on the fixed effects will generally give slightly different results).

Appendix 4: Post-fit constraint modification

Here we show how to transform model coefficients after fitting, to obtain the results that would have been obtained using alternative identifiability constraints. A simple approach is to build a version of the model matrix under the first set of constraints, $\tilde{\mathbf{X}}$, and another version under the alternative constraints, $\mathbf{X}$. Both model matrices are the whole model versions, after absorption of the constraints by reparameterization. Since the alternative constraints simply impose identifiability it is fundamental that

$$\mathbf{X}\boldsymbol{\beta} = \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}} \tag{6}$$

Now form a pivoted QR decomposition $\mathbf{QR} = \mathbf{X}$ (so the columns of $\mathbf{X}$ may be pivoted here, and this will need to be reversed later). If $\mathbf{R}$ is full rank then

$$\boldsymbol{\beta} = \mathbf{R}^{-1}\mathbf{Q}^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}} \tag{7}$$

($\boldsymbol{\beta}$ in pivoted order), and we are finished.

If $\mathbf{R}$ is not full rank (since $\mathbf{X}$ need not be), but of rank $r < p = \dim(\boldsymbol{\beta})$, then only its first r rows will be non-zero: denote these by $r \times p$ matrix $\mathbf{R}_1$. Let $\mathbf{Q}_1$ denote the corresponding first r columns of $\mathbf{Q}$. Then the r constraints imposed by (6) become

$$\mathbf{R}_1\boldsymbol{\beta} = \mathbf{Q}_1^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}. \tag{8}$$

Given these constraints the fitting process seeks to minimize the model penalty, $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}$, say. So we have a simple quadratic optimization subject to linear constraints. To proceed we require range and null space bases for the

constraints, which can be obtained by QR decomposition (no need to pivot)

$$\mathbf{R}_1^{\mathrm{T}} = \bar{\mathbf{Q}} \begin{bmatrix} \bar{\mathbf{R}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Y} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{R}} \\ \mathbf{0} \end{bmatrix}$$

where $\bar{\mathbf{R}}$ is $r \times r$ and $\begin{bmatrix} \mathbf{Y} & \mathbf{Z} \end{bmatrix}$ is just a partitioning of $\mathbf{Q}$ ($\mathbf{Y}$ is $p \times r$). Setting

$$\boldsymbol{\beta} = \begin{bmatrix} \mathbf{Y} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_Y \\ \boldsymbol{\beta}_Z \end{bmatrix}$$

then it is easy to show that the constraints (8) are satisfied by any $\boldsymbol{\beta}_Z$ if $\boldsymbol{\beta}_Y = \bar{\mathbf{R}}^{-1}\mathbf{Q}_1^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}$. So we seek any $\boldsymbol{\beta}_Z$ to minimize $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}\boldsymbol{\beta}$, which is equivalent to finding the $\boldsymbol{\beta}_Z$ minimizing $\boldsymbol{\beta}_Z^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}}\mathbf{S}\mathbf{Z}\boldsymbol{\beta}_Z$. Clearly $\boldsymbol{\beta}_Z = \mathbf{0}$ will achieve this and hence

$$\boldsymbol{\beta} = \mathbf{Y}\bar{\mathbf{R}}^{-1}\mathbf{Q}_1^{\mathrm{T}}\tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}} \tag{9}$$

is the more general version of (7) (again in pivoted order, as a result of the first QR decomposition).

Since covariance matrices also have to be transformed, it may be useful to explicitly form the matrix $\mathbf{P}$ such that $\boldsymbol{\beta} = \mathbf{P}\tilde{\boldsymbol{\beta}}$, from the definitions in (7) or (9). In that case if $\mathbf{V}_{\tilde{\beta}}$ is a covariance matrix in the fitting parameterization, $\mathbf{V}_\beta = \mathbf{P}\mathbf{V}_{\tilde{\beta}}\mathbf{P}^{\mathrm{T}}$ is the equivalent assuming the alternative constraints.

References

Bates, D and Maechler, M. (2010) lme4: Linear mixed-effects models using S4 classes. http://CRAN.R-project.org/package=lme4.

Belitz, C. and S. Lang. (2008) Simultaneous selection of variables and smoothing parameters in structured additive regression models. Computational Statistics & Data Analysis, 53(1):61-81.

Borchers, D.L., S.T. Buckland, I.G. Priede and S. Ahmadi (1997) Improving the precision of the daily egg production method using generalized additive models. Canadian Journal of fisheries and aquatic science, 54: 2727-2742.

Breslow, N.E. & D.G. Clayton (1993) Approximate inference in generalized linear mixed models. Journal of the American Statistical Association, 88: 9-25.

Davison, A.C. (2003) *Statistical Models*. Cambridge.

Eilers P.H.C. and B.D. Marx. (1996) Flexible smoothing with B-splines and penalties. Statistical Science, 11(2):89-102.

Eilers P.H.C. (1999) discussion of Verbyla, A.P., B.R. Cullis, M.G. Kenward and S.J. Welham (1999) The Analysis of Designed Experiments and Longitudinal Data by Using Smoothing Splines. Journal of the Royal Statistical Society, Series C, 48(3):307-308.

Eilers, P. H. C. and Marx, B. D. (2003) Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. Chemometrics and intelligent laboratory systems, 66: 159-174.

Fahrmeir, L., T. Kneib & S. Lang (2004) Penalized structured additive regression for space time data: A Bayesian perspective. Statistica Sinica, 14: 731-761.

Gu, C. (2002) *Smoothing Spline ANOVA Models*. Springer.

Gu. C. & Y-J Kim (2002) Penalized likelihood regression: general formulation and efficient approximation. The Canadian Journal of Statistics, 30(4): 619- 628.

Harville, D.A. (1997) *Matrix Algebra From a Statisticians Perspective*. Springer.

Hastie, T. & R. Tibshirani (1986) Generalized additive models (with discussion). Statistical Science, 1: 297-318.

Kim, Y. J. and Gu C. (2004) Smoothing spline Gaussian regression: more scalable computation via efficient approximation. Journal of the Royal Statistical Society, Series B, 66: 337-356.

Kimeldorf, G and G. Wahba (1970) A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. Annals of Mathematical Statistics, 41: 495-502.

Lee, D-J and M. Durbán (2011) P-spline ANOVA type interaction models for spatio-temporal smoothing. Statistical Modelling, 11(1): 49-69.

Lin X. and Zhang, D. (1999). Inference in generalized additive mixed models using smoothing splines. Journal of the Royal Statistical Society, Series B, 61: 381-400.

Parker, R. and J. Rice (1985) Discussion of Silverman (1985) Journal of the Royal Statistical Society, Series B, 47(1): 41-42.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-PLUS*. Springer.

R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, `www.R-project.org`.

Ruppert, D., M.P. Wand & R.J. Carroll (2003) *Semiparametric Regression*. Cambridge.

Silverman, B.W. (1985) Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). Journal of the Royal Statistical Society, Series B, 47: 1-53.

Verbyla, A.P., B.R. Cullis, M.G. Kenward and S.J. Welham (1999) The Analysis of Designed Experiments and Longitudinal Data by Using Smoothing Splines. Journal of the Royal Statistical Society, Series C, 48(3):269-311.

Wahba, G. (1980) Spline bases, regularization and generalized cross validation for solving approximation problems with large quantities of noisy data. in E. Cheney (ed) Approximation Theory III Academic Press, London.

Wahba, G. (1990) *Spline Models for Observational Data*. SIAM, Philadelphia .

Wood, S.N. (2004) Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models. Journal of the American Statistical Association, 99: 673-686.

Wood, S.N. (2006a) Low-Rank Scale-Invariant Tensor Product Smooths for Generalized Additive Mixed Models. Biometrics, 62(4): 1025-1036.

Wood, S.N. (2006b) *Generalized additive models: An introduction with R*. Taylor & Francis/ CRC Press.

Wood, S.N. (2008) Fast stable direct fitting and smoothness selection for Generalized Additive Models. Journal of the Royal Statistical Society, Series B, 70(3): 495-518.

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society, Series B, 73(1): 3-36.